

**REMARKS**

Please reconsider the application in view of the above amendments and the following remarks. Applicants thank the Examiner for carefully considering this application.

**Disposition of Claims**

Claims 1-8 were 10-23 pending in this application. Claims 2-4, 7-8, 10-12, 14, and 18-20 have been cancelled in this reply without prejudice or disclaimer. Claims 24-25 have been added in this reply. Accordingly, claims 1, 5-6, 13, 15-17, and 21-25 remain pending in this application. Claims 1, 13, and 23 are independent. The remaining claims depend, directly or indirectly, from independent claims 1 and 13.

**Claim Amendments**

New dependent claims 24 and 25 have been added in this reply and include the subject matter of claims 15 and 16 respectively. Claims 1, 5, 13, and 23 have been amended in this reply to further clarify embodiments of the invention. Support for these amendments may be found, for example, in paragraphs [0018]-[0024] of the originally filed specification. Applicants submit that no new matter has been added by way of these claim amendments.

**Rejections under 35 U.S.C. § 102**

“A claim is anticipated only if *each and every element* as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.” *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631 (Fed. Cir. 1987) (emphasis added). Further, “[t]he

identical invention must be shown in as complete detail as is contained in the claim.” *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236 (Fed. Cir. 1989).

Claim 23 stands rejected under 35 U.S.C. § 102 as being anticipated by U.S. Patent No. 6,728,955 (“Berry”). To the extent that this rejection applies to the amended claims, the rejection is respectfully traversed.

Amended independent claim 23 recites, in part, a processor configured to “*load*, in response to triggering a hook in an initialization file associated with the instrumented application, a helper action into the kernel level for use by a tracing framework, *wherein the helper action is* a stored procedure generated using an implementation specific detail associated with the instrumented application for obtaining a stack trace of the instrumented process, and wherein the helper action is linked to the initialization file associated with the instrumented application.” Independent claim 23 further requires, in part, for the processor to be configured to “*register* the helper action with the tracing framework.” Independent claim 23 further requires, in part, that the processor executes the tracing framework, with the tracing framework configured, in part, to “*determine* [...] whether the helper action is associated with the probe based on the registration of the helper action with the tracing framework” and to “*perform* the helper action to obtain the stack trace of the instrumented process when the helper action is associated with the probe.” As such, all acts pertaining to the helper action (*i.e.*, loading, registering, performing, etc.) are qualified at minimum by the limitation that the helper action is “is a stored procedure generated using an implementation specific detail associated with the instrumented application for obtaining a stack trace of the instrumented process.”

In contrast, Berry does not disclose the use of a helper action. Berry primarily discloses two general types of metrics that are recorded as a result of tracing with a profiling process. A first type pertains to time-related statistics with regards to each of the routines found in a call stack. *See* Berry, column 14 line 20 – column 17 line 10. A second type pertains to increasing metrics, such as number of I/O actions, number of bytecodes executed, amount of memory allocated, number of cache misses, time elapsed, number of memory allocations/deallocations, and number of bytes interpreted. *See* Berry, column 22 line 46 – column 23 line 13. As such, Applicants assert that these metrics are general in nature and can be drawn from *any* system process let alone a particular process with which the profiling process corresponds. Further, the collection of these metrics by a profiling process does *not* require the profiling process to possess any knowledge of implementation specific details associated with an instrumented application. Accordingly, the profiling process disclosed in Berry is more accurately characterized as generic tracing rather than a stored procedure generated using an implementation specific detail associated with the instrumented application for obtaining a stack trace of the instrumented process. *See* paragraph [0022] of the Specification. For that reason, Applicants assert that a profiling process disclosed Berry cannot be characterized as a helper action.

Moreover, Applicants assert that Berry does not disclose a helper action that (i) is “linked to the initialization file associated with the instrumented application” or (ii) loaded, in response to triggering a hook in an initialization file associated with the instrumented application, into the kernel level for use by a tracing framework. An initialization file “contains routines to be executed when the instrumented application object file is loaded into the kernel.” *See* paragraph [0017] of the

Specification.<sup>1</sup> Applicants assert that, at best, Berry discloses a profiler maintained in a dynamic link library (DLL). *See* Berry, column 25, lines 3-7. The routines in a DLL are executed when called by an executing process. In contrast, the routines for the initialization file are executed in response to the loading of the instrumented application with which the initialization file is associated. In view of this, Applicants assert that Berry does not disclose an initialization file but merely a DLL and, as such, does not disclose the above-numbered limitations (i) and (ii) of independent claim 23.

For at least these reasons, profiling a system disclosed in Berry cannot be equated to the use of a helper action as recited by the pending claims. Accordingly, Berry fails to disclose all the limitations of independent claim 23. Withdrawal of this rejection is respectfully requested.

### **Rejections under 35 U.S.C. § 103**

MPEP §2143 states that “[t]he key to supporting any rejection under 35 U.S.C. 103 is the clear articulation of the reason(s) why the claimed invention would have been obvious.” The Supreme Court in *KSR International Co. v. Teleflex Inc.*, 127 S.Ct. 1727, 1739, 75 U.S.L.W. 4289 (2007) noted that the analysis supporting a rejection under 35 U.S.C. 103 should be made explicit. *See*, MPEP §2143.

In particular, the Examiner “must articulate the following: (1) a finding that the prior art included each element claimed, although not necessarily in a single prior art reference, with the only difference between the claimed invention and the prior art being the lack of actual combination of

---

<sup>1</sup> Applicants note that while the claims do not explicitly recite a definition of an initialization file. The Examiner is required to review the terms in the claims in view of the specification. *See Phillips v. AWH Corp.*, 415 F.3d 1303, 75 USPQ2d 1321 (Fed. Cir. 2005) (*en banc*)

the elements in a single prior art reference; ..." MPEP § 2143(A). Applicants assert that the prior art, whether viewed separately or in combination, fails to teach or suggest all the limitations of the pending independent claims.

Claims 1-8 and 10-22 stand rejected under 35 U.S.C § 103 as being unpatentable over John Murayama, "Performance Profiling Using TNF", July 2001, pg. 1-6 ("Murayama") in view of Berry. Claims 2-4, 7-8, 10-12, 14, and 18-20 have been cancelled by this reply. Accordingly, this rejection is moot with respect to the cancelled claims. To the extent that this rejection applies to the pending claims, the rejection is respectfully traversed.

Amended independent claim 1 is directed to a method for tracing an instrumented application. More specifically, independent claim 1, recites in part: (i) loading the instrumented application comprising a probe into a kernel level to obtain a corresponding instrumented process; (ii) triggering, after loading the instrumented application, a hook in an initialization file associated with the instrumented application to load a helper action into a kernel level for use by a tracing framework; (iii) registering the helper action with the tracing framework; (iv) tracing the instrumented process using the tracing framework, wherein tracing comprises triggering the probe in the instrumented process; (v) determining, after triggering the probe, whether the helper action is associated with the probe based on the registration of the helper action with the tracing framework; (vi) obtaining the helper action when the helper action is associated with the probe; and (vii) performing the helper action to obtain the stack trace of the instrumented process when the helper action is associated with the probe.

Claim 1 defines the helper action as "a stored procedure generated using an implementation specific detail associated with the instrumented application for obtaining a stack trace of the

instrumented process” and qualifies the term by adding that “the helper action is linked to the initialization file.” In view of the amended claim and the specification, a helper action is specific to the instrumented process or a set of instrumented processes. *See* Specification, paragraphs [0016], [0018], and [0022]. In contrast, a generic tracing action is not specific to an instrumented process. *See* Specification, paragraph [0022].

In view of this, Applicants assert Murayama does not disclose the use of a helper action in tracing an instrumented application. With regards to instrumenting a target, the prior art discloses inserting TNF probes into source code which, at best, provide a generic logging functionality that is adjustable via parameterization by a developer. *See* Murayama, page 2: ‘Instrumenting the Target ...’, Example 1, and ‘Inserting Probes’ (“*A TNF probe is a parameterized macro that allows a user to specify up to 5 argument values for the probe trace to record.*”). As such, under Murayama, the macro function invoked by the TNF probe is capable of use with any number of instrumented applications and is therefore not “generated using an implementation specific detail associated with the instrumented application.” Further, the fact that a developer can parameterize the calling of the macro function does not mean that the macro function has been generated using implementation specific details associated with the instrumented application for the reason that it is the same generic, all-purpose macro function code that is being invoked in each instance. For at least these reasons, instrumenting the target using a TNF probe, as disclosed by Murayama, achieves system profiling through *merely generic* tracing actions and *not* through use of a helper action as claimed.

Because Murayama does not disclose the use of a helper action recited in the amended claims, it then follows that Murayama fails to teach or suggest other limitations cited in claim 1: (i) linking the helper action to the initialization file associated with the instrumented application; (ii)

triggering, after loading the instrumented application, a hook in an initialization file associated with the instrumented application to load a helper action into a kernel level for use by a tracing framework; (iii) registering the helper action with the tracing framework; (v) determining, after triggering the probe, whether the helper action is associated with the probe based on the registration of the helper action with the tracing framework; (vi) obtaining the helper action when the helper action is associated with the probe; and (vii) performing the helper action to obtain the stack trace of the instrumented process when the helper action is associated with the probe.

*Arguing in the alternative* that tracing an instrumented application using a TNF probe *can be* equated to tracing an instrumented application using a helper action, Murayama *still fails to teach or suggest* the registration of the helper action with the tracing framework. The Examiner relies upon use of interposition libraries under Murayama as evidence of registering a helper action. *See* page 2 of the Action. To this end, Murayama discloses the redirection of function calls to thread library wrapper functions which are equipped to effectively enclose the function call between a pair of TNF probes. *See* Murayama, page 3: Example 3 and ‘Interposition Libraries’. As such, this feature *merely constitutes* instrumenting a target by using the thread layer as a proxy for the source code layer. This is particularly useful in instances where the source code is not available for altering or recompiling. *See* Murayama, page 3: ‘Interposition Libraries.’ Because the cited-to use of interposition libraries is still *merely a manner of instrumenting* a target, it cannot be characterized as registering a helper action with a tracing framework which is a *distinctly separate limitation* performed under the pending claims *after an application has already been instrumented*. In contrast, the registration of a helper action with a tracing framework enables the tracing framework

to associate the registered helper action with a triggered probe in an instrumented process. *See* Specification, paragraph [0022].

Moreover, Berry does teach or suggest that which Murayama lacks. As argued previously in “Rejections under 35 U.S.C. § 102”, *see* pages 7-8 of this Response, Berry does not contemplate the use of “a stored procedure generated using implementation specific details associated with the instrumented application for obtaining a stack trace of the instrumented process.” Rather, Berry can be characterized merely as disclosing a generic tracing functionality that is not specific to any particular instrumented application. For these reasons, profiling a system as disclosed by Berry cannot be equated to the use of a helper action in view of the amended claims. As such, Berry fails to teach or suggest all the limitations of amended independent claim 1.

In view of the above, independent claim 1 is patentable over Murayama and Berry. Further, claim 13 includes at least the same patentable limitations as independent claim 1 and accordingly, claim 13 is patentable over Murayama and Berry for at least the same reasons as discussed above with respect to independent claim 1. Dependent claims are patentable for at least the same reasons. Accordingly, withdrawal of this rejection is respectfully requested.

## **New Claims**

New claims 24-25 have been added in this reply. As discussed above, amended independent claim 1 is patentable over the cited prior art. Further, claims 24-25 depend from claim 1. Thus, claims 24-25 are allowable for at least the same reasons as independent claim 1. Accordingly, a favorable action in the form of a Notice of Allowability is respectfully requested.



**Conclusion**

Applicants believe this reply is fully responsive to all outstanding issues and places this application in condition for allowance. If this belief is incorrect, or other issues arise, the Examiner is encouraged to contact the undersigned or his associates at the telephone number listed below. Please apply any charges not covered, or any credits, to Deposit Account 50-0591 (Reference Number 03226/369001; SUN040527).

Dated: February 4, 2009

Respectfully submitted,

By           /Robert P. Lord/            
Robert P. Lord  
Registration No.: 46,479  
OSHA · LIANG LLP  
909 Fannin Street, Suite 3500  
Houston, Texas 77010  
(713) 228-8600  
(713) 228-8778 (Fax)  
Attorney for Applicants